Winter 12-2-2011

An Intelligent Online Shopping Guide Based On Product Review Mining

Heng Tang

An Intelligent Online Shopping Guide Based on Product Review Mining

Heng Tang, University of Macau, hengtang@umac.mo

ABSTRACT

This position paper describes an on-going work on a novel recommendation framework for assisting online shoppers in choosing the most desired products, in accordance with requirements input in natural language. Existing feature-based Shopping Guidance Systems fail when the customer lacks domain expertise. This framework enables the customer to use natural language in the query text to retrieve preferred products interactively. In addition, it is intelligent enough to allow a customer to use objective and subjective terms when querying, or even the purpose of purchase, to screen out the expected products.

Keywords: Intelligent Shopping Guidance; Natural Language Processing; Case-based Reasoning; Ontology; Naïve Bayesian

INTRODUCTION

Background and Motivation

Sophie wants to buy a new digital camera from Amazon for her graduation trip to Paris: a portable, inexpensive camera able to shoot nice photos in museums where she is going to spend a lot of time. However, since Sophie is a pure outsider to digital photography and electronic products, she Googles key words "digital camera for museum" and browses the results. The first hit is a list of cameras by a particular brand, followed by a number of webpages offering tips for taking photos in museums with terms that are Greek to her. Then she visits a professional digital camera review website (dpreview.com). It provides a wonderful tool named "buying guide" which helps users filter out the most wanted camera from the camera database. However, Sophie is totally lost when she is asked to choose features like zoom range, ISO range, prime lens, sensor size, and exposure bracketing, etc. After 1 hour's frustrating searching and browsing, she gives up and drives out to the Bestbuy store, hoping she can get some advice from the shop assistant.

Sophie's experience is typical in the context of online purchasing. Actually, only a small proportion of online shoppers can be termed as "prosumers" with adequate domain knowledge who are able to locate the most wanted product by using search engines, browsing professional online discussion board and studying products' features [1, 2] such as ISO range or constant aperture. In contrast, there is a large number of consumers who can do nothing but just describe the desired *functions* or *requirements* of the product (e.g., low noise and macro photo, etc.). Moreover, the least professional users are perhaps only able to describe their *goals* of purchasing (e.g., taking picture of pet or flowers, "when skin diving" or in museums, etc.). As evident from Sophie's story, conventional purchasing guide systems such as the one Sophie tried cannot provide adequate guidance to non-prosumers to find the target product effectively and efficiently.

Another category of intelligent software, called Recommender Systems (RS), also strives to recommend the most needed product to the users [3]. Collaborative filtering approach and content-based approach are two most widely used recommendation methods, and the former has been reported to be highly effective and efficient for intelligent recommendation making [3, 4]. Conventional CF is based on assumption that either the user has prior knowledge and/or interest in items similar to the target product (item-based approach), or like-minded users have rated the targeted product (user-based approach). Conventional CF, therefore, may fail when these assumptions are not satisfied [5, 6]. Indeed, when people want to purchase something they know little about (e.g., a digital camera), they most likely seek suggestions from someone with domain expertise [7, 8]. More and more people are choosing to read product reviews on the internet, online discussion boards, or e-commerce websites. Empirical evidence has shown that consumers tend to believe opinions in online review articles more than commercial advertisements [9]. In this regard, some researchers have recently applied opinion mining to construct knowledge base of products, in an attempt to suggest the right product to customers using various recommendation methods [2, 10, 11].

Automatic recommendation mechanisms based on opinion-mining techniques constitute a plausible way of providing intelligent shopping guidance. The underlying assumption of this approach, however, is that the customer has the ability to specify features of the product sought to be purchased, which is an obvious obstacle when the customer's expertise is insufficient. A typical query that a user could state is, for example, "an 8 mega pixels' digit camera with very long battery life, for taking photo in museum." First, we notice that "8 mega pixels" is associated with the value of the sensor feature of a digital camera, where the association can be derived from the specifications record in the product database. Second, "very long (battery life)" is a descriptive expression of battery life because the user is unable to express the specific measure of battery sustainability, such as "mAh". Third, since the customer has no idea about what kind of camera is suitable for the purpose "for taking photo in museum", this purpose of purchasing may be directly specified in the query. This example reveals the inability of conventional feature-based recommendation approaches to serve shoppers with low domain expertise.

In this paper, we propose a framework for assisting shoppers in choosing the most favorable products, without requiring them to have much domain knowledge. Especially, users are allowed to describe their requirements or objectives of purchasing in natural language. The coded requirements are then delivered to the inference engine for discovering the most matched products. The challenges of this research are that 1) how can the requirements be coded and understood by

the system, given that the level of expertise of customers in the product domain varies; 2) how can the knowledge base of the target product be constructed with as little human effort as possible; and 3) how do we design the matching or recommendation algorithm that identifies products most pertinent to the customer's needs.

The Overall Framework and Contributions

In this study, we propose a generic framework to analyze product reviews and provide recommendations for shoppers, as illustrated in Figure 1. The framework consists of two major phases, product Knowledge Base (KB) Construction and Matching. In the first phase, review texts are processed in order to extract features and opinions, using the Natural Language Processing (NLP) module. In this process, the product database and ontologies defined on the domain are needed for retrieving structured product information and understanding the semantics in the review texts. The output of this phase is the product knowledge base for supporting future recommendations. In the second phase, when a customer's query is received, the NLP Module analyzes and encodes the query text and then compares it with products saved in the product KB by the Matching Engine. The output of this phase is the best fit products for the customer's reference.

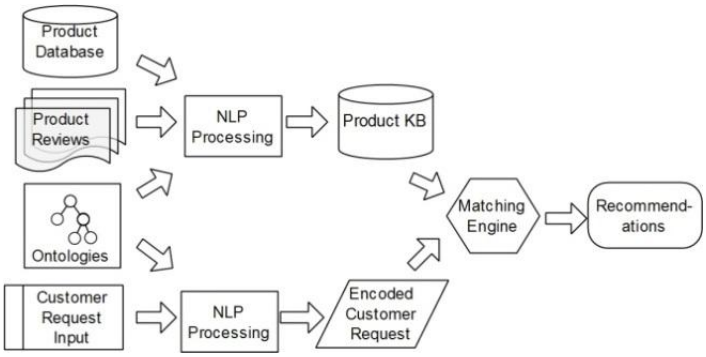

Figure 1. The Overall Framework

In sum, the main contributions of this paper include: 1) Presenting a case-based product recommendation framework with detailed procedures to assist online shoppers; 2) Differentiating and proposing three types of requests in user query text; and 3) Proposing various distance measures to be used for estimating the similarity between the query and the products. This framework can be applied to assist shoppers in discovering their desired products, and hereafter in this paper, we use the e-shopping guidance as a work-through example to demonstrate the efficacy of the proposed methods. This framework, however, is generalizable to many other applications where items are to be recommended to users who are unable to describe their requirements precisely.

The remainder of the paper is structured as follows. In Section 2, a brief review of relevant literature is provided. In Section 3, three types of user requests are explicitly differentiated and the NLP module to process query text and review corpus is introduced. The construction of the ontologies is also discussed in this section. In Section 4, the matching engine is elaborated, with the formulation of distance measures. Section 5 describes the matching algorithm. The case-based recommendation process is presented in Section 6. Section 7 summarizes the paper and outlines future research directions.

BACKGROUND AND LITERATURE

Recently, systems to suggest the right products to the customer, normally called Recommender Systems or Recommendation Systems (RS), have drawn much attention from scholars. Recommender Systems are defined as intelligent programs which strive to identify products of the most interest to the users, given their historical interests or actions [3]. A recommender system attempts to predict the 'rating' that a user might assign a product by examining some specific characteristics in its profile. These characteristics can be related to the product and the user (the Content-based approach), the user's social environment (the Collaborative Filtering approach) or both (the hybrid approaches) [3]. A user's profile is normally generated by analyzing previous rating information which could be either explicit or implicit [4]. Recommendation systems, especially, have been extensively utilized in e-commerce domains for shopping aid and product recommendation [4, 12]. Specifically, in-depth research on Collaborative Filtering (CF) has been conducted by researchers. CF does not need explicit description of content generally required in the content-based approach for calculation of the similarity between an item and a user's interests. Instead, CF provides recommendations according to user preferences by maintaining users' purchasing record for identifying users with similar tastes. Thus products liked by a user can be introduced to other people of the same kind. However, CF-based systems are known to suffer cold-start problem (new user) and sparsity [3, 5].

It has been recognized that lately, the review or discussion of products on online forums and e-commerce websites has become an important source of information about opinions about a product [13-15]. There is evidence showing that opinions contained in online reviews may significantly affect customers' purchase decisions [9], which can be exploited by intelligent systems to provide better recommendations. In many e-commerce websites or product review discussion boards, explicit ranking scores of different products are available with the review text, normally on 5-point Likert scale. However, making recommendation simply based on the ranking may be problematic since readers' personal tastes may differ from those of the reviewers [16]. For example, a user may be fond of an ultra-compact digit camera regardless of the photo quality, while the reviewers may place more weight on the latter. As such, the reader will have to go through a large amount of review articles, try to digest many unfamiliar terminologies, compare many choices, and make the final decision on its own. Consequently, data mining and machine learning techniques, coupled with natural language processing approaches for extracting product ranking and other valuable information from product review texts have come to be referred to as Opinion Mining [17, 18]. For example, in [11], an intelligent recommendation approach is proposed, which is based on scores discovered from online reviews.

To have recommender systems with a deeper

understanding of customer reviews, researchers apply feature extraction techniques in order to automatically identify the keywords of features or opinions. A number of well-established approaches in NLP can be utilized for this purpose. For example, part-of-speech (POS) tagging tools [19], which can be used to identify POS of words (e.g., adjective or adverb, etc.) in the review text. Some studies consider both product features and subjective terms when comparing products. Notably, Red Opal [2], a recommender system based on opinion mining, explores online customer reviews in order to identify product features and automatically score products according to their features. Hence the most suitable product can be recommended by matching products and features specified by the customer. Opinion mining techniques are also utilized for automatic differentiation of sentimental orientation (recommended or not) towards an item expressed in the text [20], which is especially useful in supporting intelligent recommendation. Notice that dealing with low-quality review corpus is out of the scope of this paper; we assume that all review documents are high-quality (i.e. without noisy data or spam reviews). Readers interested in detection of noisy or spoof reviews may refer to [21-24].

PROCESSING PRODUCT REVIEW AND QUERY

In the overall framework, either the processing of product information to establish the product KB in the first phase or the processing and encoding of the customer query text in the second phase are essentially based on NLP techniques. Hence these two tasks are introduced together in this section.

The goal of guiding a customer to find the favorable product can be achieved by matching the query text with information of products stored in the product knowledge base. In this research, a customer's requirements defined in terms of the product's features, performance parameter and usage context, etc., are referred to as a *request*. In order to understand the customer's query without requiring precise specifications of product features, we explicitly differentiate three types of requests, i.e. Objective Request (OR), Subjective Request (SR) and Usage Request (UR). An *Objective Request* accurately and objectively describes the factual information about the product. For example, in the request "14 megapixel sensor", "14 megapixel" is the objective term used to describe the feature "sensor". This type of terms are highly domain-specific. The more ORs the query text from the customer has, in general, the higher is the expertise level. *Subjective Request* describes product features with subjective words, e.g., "*high resolution*" or "*portable"*, which are generally provided by customers who are unable to name the precise requirement on a feature. SR is characterized by adjectives and adverbs in the text. This type of request is also domain-specific in general [25]. *Usage Requests* are normally from novices who are merely able to describe the usage of the product or the purpose of purchasing, for instance, "…a digital camera to take nice photo in museum" or "…taking portrait photos".

Given the three types of customer requests defined above, the tasks of constructing the product knowledge base include 1) extracting and summarizing product information from the product database, online reviews or other product information sources; and 2) encoding obtained product information and storing it in the knowledge base in order to allow mapping between user requests (OR, SR, or UR) and product items. The mapping is used for comparison, to find the best match. In this study, information sources for establishing the product knowledge base include the product database and product reviews corpus. The former generally provides precise and objective descriptions of products, while product information in the latter is indirect, to be derived using various NLP and opinion mining approaches.

A. Encoding Product Knowledge

In this research, the definition of Feature-Opinion Pair for movie reviews mining [26] is modified to formalize descriptions of features in either product reviews or user queries.

DEFINITION (Product Feature). A product feature is an attibute of product (such as "zoom range"). It could appear in the product database, product review text, or user query text, etc.

DEFINITION (Feature Value). A feature value is the actual value related to the corresponding product feature.

A sentence in a product review can thus be represented as the set of Feature-Value pairs. For example, a sentence in a product review "Its maximum ISO is up to 6400" can be denoted by the pair ("maximum ISO", "6400"), while a sentence in a subjective query request "…a camera producing high resolution pictures" can be represented by the pair ("Resolution", "High"). Notice that the feature or value can sometimes be absent in a Feature-Value pair, which can be considered as a pair with implicit feature or value. Likewise, sentences in the query text from a customer can be encoded into a set of Feature-Value pairs. Each pair in the query is called a *request* in this paper. Specifically, a Usage Pair in the query can be represented by assigning a values of "1" to the feature, for example ("scuba diving", 1), meaning the camera must be suitable for the usage context "scuba diving".

Following the definition of OR, SR and UR, we have the corresponding concepts *Objective Pair*, *Subjective Pair* and *Usage Pair*, which are Feature-Value pairs whose value domain is objective, subjective and usage related, respectively. An Objective Pair can be extracted from a sentence in an Objective Query or the product database which provide precise description of the products. A Subjective Pair, in general, can be extracted from a sentence in either a Subjective Query or a product review text. Likewise, a Usage Pair can also be extracted from a sentence of either a Subjective Query or a product review text.

With the representation of feature-value pair, either the product information or the customer query can be represented by a set of pairs. Assuming $F = \{f_1, f_2, \dots, f_m\}$ is the set of all available features, product *p* can be formalized as a vector of feature-value pairs defined on a subset of *F*, denoted as $p = [PP_1, \dots, PP_{kp}]$, where $PP_j = (f_{jp}, v_{jp}), 1 \le j \le k_p$, is a product feature-value pair (objective, subjective or usage). Similarly, a customer query can be represented by $q = [PQ_1, \dots, PQ_{kq}]$, where $PQ_j = (f_{jq}, v_{jq}), 1 \le j \le k_q$ is a request feature-value pair (objective, subjective or usage). The similarity between a query and a product, therefore, can

be estimated by accounting for the distance between the corresponding feature values while their feature names appear in both vectors. Details about comparison of vectors are introduced in Sections 4 and 5.

B. The NLP Modules

Product databases offer well-organized and detailed descriptions of product features, which become an important information source of the product knowledge base. Capitalizing knowledge hidden in review documents, in contrast, is much more challenging since product reviews are normally free text based. In this study, NLP and text mining techniques are used in order to extract the needed information from the review corpus (i.e. the NLP module in figure 1). Given a collection of review documents about a type of product collected from the web, as well as the database of product, the data source of the NLP module is ready for the process. The procedure to extract a product KB includes the following 3 steps:

1) Preprocessing. The list of all products (digital cameras) is generated; each product is assigned a unique identity and its associated review articles are saved in the reviews corpus.

2) Annotation. This step parses review texts and annotates elements with tags. Most techniques used in this step are based on a few well-developed and widely used NLP techniques [27, 28]. It includes tokenization (breaking down the original review text into tokens such as punctuations, numbers, spaces and words, etc.), Named Entity detection (i.e. identifying entity names such as "prime lens" according to the predefined list or ontologies), Sentence-splitting (segmenting the text into sentences), and POS-tagging (based on the context and definition of a word, tagging it as corresponding to a particular part-of-speech, i.e. nouns, verbs, adjectives and adverbs). Natural Language Toolkit (NLTK [27]), an open source library, can be utilized to implement the above process.

3) Feature-Value pair identification. Based on the output of the previous step, i.e. review texts annotated with various tags, this step identifies the Feature-Value pairs by extracting features and their corresponding values, according to some predefined rules (such as fixed syntactic phrases, etc.) or ontologies. For instance, a feature of a product is generally a noun or noun phrases, which can be retrieved from the product feature lexicon generated from the product database. The feature value can be either adjective/adverb phrases or implicit (in the case of a usage pair). Notice that in some reviews, some Feature-Value pairs can be implicit. For instance, "I decided to sell out my original Nikon DSLR right after testing this tiny camera." Since understanding this type of review opinion is very difficult, if not impossible, the proposed framework only deals with pairs expressed in an explicit way. The output of this step is product information encoded with the form of the vector, which consists of the ID of the product and a number of pairs describing its features. For example, "Sony W70": [("resolution", "10mp"), ("portable", "good"), …].

Throughout the processes of Step 2 and 3, in addition, the support of ontologies is needed in order to allow identification of named entities with resembling semantics. For example, ("sensor", "large") and ("CMOS", "big size") are semantically equivalent indeed.

C. Developing Ontologies

Automatic extraction of feature names and values from review texts requires the system to understand text written in natural language, which has been known as a big challenge. Ontology, recognized as a powerful tool for understanding and capitalizing domain knowledge [29], is incorporated in this research. Ontology can help the proposed framework form an unambiguous understanding of semantics of user reviews which, in general, is unstructured information [30]. It is believed that the semantic representation of lexica plays a key role in full utilization of hidden information in the product review. It can eliminate ambiguity and help fix the imprecision or incompletion in the review. Web ontology languages (such as OWL) can help interpret various contextual concepts related to different products.

In general, three paradigms are widely adopted for the construction of an ontology, i.e. bottom-up, top-down and hybrid approach [31]. The top-down paradigm starts with existing domain resources (such as taxonomies) and heuristic knowledge, and then increasingly provides more details afterwards. The bottom-up paradigm, on the contrary, starts from the raw documents, attempting to identify and extract lexica for the ontology. A hybrid paradigm starts from the concepts, construction and raw document extraction at the same time, and tries to establish the mappings between the ontological levels. At the beginning, words baseline occurrence rates [2] can be utilized to identify terms used to initialize the ontologies. Although much research has been done on automatic ontology construction [32], manpower is still needed for optimization and validation, so that a practical and usable ontology can be established. Several ontologies developed in this framework are as follows.

- Product ontology

The product ontology is the ontology developed for the specific product domain to support drawing of inferences from among feature terms. The hybrid paradigm can be adopted to build the ontology for the specific product. In the top-down stage, product ontologies can be constructed by capitalizing meta information of products available in the product databases, e-Commerce, or product review websites. For example, the DPreview (www.dpreview.com) website provides an updated and comprehensive review database of products related to digital photography, such as digital cameras. In many product databases, specifications such as sensor, ISO range, metering, focus mode, dimension and weight, etc., are also available in a well-structured format. Those descriptions of a digital camera constitute the features that the customer may consider and compare when about to purchase a new camera.

The output of the top-down stage is the preliminary ontology of a product, which can be used to guide the bottom-up stage. Content of review documents can be analyzed to help identify the taxonomies, synonyms and so on, from the corpus [33]. This stage further consolidates the product ontology by inserting, deleting and refining the properties in its draft version.

- Descriptive ontology

Descriptive ontology maintains semantics of subjective terms and their associations. The development of descriptive ontology also follows a hybrid approach. In the top-down stage, synsets in WordNet [34] can be utilized to generate a primitive descriptive ontology. The bottom-up stage can be

carried out along with that of product ontology since the descriptive terms are largely dependent on the specific product domain as well. For example, the feature "range of color variations" is associated with descriptive terms "wide" and "narrow".

- Sentiment ontology

Sentiment ontology is the ontology for understanding and references about sentimental terms. Top-down approach can be adopted to develop the sentiment ontology. The foundation of this ontology is SentiWordNet [35], a lexical resource widely used for processing natural language for the understanding of sentimental terms and proven to be effective. In SentiWordNet, polarity information is quantified on the basis of the lexica in WordNet, using linguistic and statistic classifiers. And a synset in SentiWordNet is associated with three polarity scores (positivity, negativity and objectivity) and the sum of the three maintains 1. For instance, the triplet (0, .75, .25) (positivity, negativity, objectivity) is assigned to the term "poor". Sentiment ontology can be utilized to measure the orientation of opinions towards a product or its features for deriving the implicit ranking information.

- Usage ontology

In a product review, usage ontology models the usage terms describing situations with regard to the environment in which the product performs well. For example, a review stating "especially suitable for night shot" implies the product (camera) is evaluated high in the usage context "night shot". In a customer's query text, similarly, context information indicates the purpose of purchase or the environment in which the product is used. For example, "…a compact camera for taking photo underwater", in which "underwater" defines the usage environment. A well-defined usage ontology plays a key role in allowing utilization of the usage information hidden in the reviews corpus and comprehending the usage request in the query text in an unambiguous way. For example, "for outdoor", in terms of the usage of a camera, semantically resembles to "for hiking".

THE MATCHING ENGINE

With the representation of feature-value pair, either the product information or the customer query can be represented by a set of pairs. The similarity between a query and a product, therefore, can be estimated by the synthesis of the distance of the corresponding pairs. Various distance measures can be adopted for the synthesis. The following paragraphs introduce the distance between three types of pairs.

i) Distance between Objective Pairs

For most products, precise description of features is available in product databases provided by the manufacturer or other sources. Product information in these repositories is generally structured or semi-structured, and hence query with customers' objective query is straightforward. Let $\mathrm{PP} = (\mathrm{f}, \mathrm{v_p})$ be an Objective Pair about a product *p* and $\mathrm{PQ} = (\mathrm{f}, \mathrm{v_q})$ be an Objective Pair from a query *q*, *PP* and *PQ* share the same product feature *f*. The distance between objective pairs *PP* and *PQ* is the distance between two feature values $\mathrm{v_p}$ and $\mathrm{v_q}$, denoted as

$$dist(PP, PQ) = d(v_p, v_q).$$

When the feature is numerical, e.g., "the LCD size", the distance measure *d*( ) can be simply calculated as the arithmetic distance between two numeric values standardized into the interval [0,1]. Otherwise, when the feature is categorical or textual (e.g., "with/without viewfinder"), one easy way to quantify *d*( ) is to compare whether the two strings are (approximately) equivalent.

Notably, the semantic equivalence should be considered since product information may be collected from various sources. For example, "sensor size" versus "CMOS size", and "12 megapixel" versus "12 MP". Therefore, the definition of an ontology on the product domain is needed, and thus a graph-based approaching accounting of the traversed distance along weighted arcs in the sematic network can be used to calculate the semantic distance [36].

ii) Distance between Subjective Pairs

Unlike the Objective Pairs, directly calculating the distance between values of two subjective pairs is difficult, due to the high complexity of human language in describing subjectivity, and imprecision. Lately, a very effective solution to this issue is WordNet [34]. WordNet is a lexical database grouping English words into sets of synonyms and providing the semantic relations between these sets. For example, ("LCD size", "*big*") and ("LCD size", "*large*") can be recognized as synonyms pairs.

In the proposed framework, WordNet is used to calculate the semantic distance [37] between values of two Subjective Pairs. Let $\mathrm{PP} = (\mathrm{f}, \mathrm{v_p})$ be a Subjective Pair about a product *p* and $\mathrm{PQ} = (\mathrm{f}, \mathrm{v_q})$ be a Subjective Pair from a query *q*, *PP* and *PQ* share the same product feature *f*. Since subjective product information is extracted from user reviews corpus, the distance $\mathrm{dist}(\mathrm{PP}, \mathrm{PQ})$ should be considered as the average distance between *PQ* and all pairs related to feature *f* of product *p* in the reviews corpus. Therefore:

Let $\mathrm{R_p} = \{\mathrm{r_1}, \mathrm{r_2}, \dots, \mathrm{r_n}\}$ be a set of reviews of the same product *p*, and in each review $\mathrm{r_i}, 1 \le \mathrm{i} \le \mathrm{n}$, assuming $\mathrm{S_i} = \{\mathrm{s_{i1}}, \mathrm{s_{i2}}, \dots, \mathrm{s_{im}}\}$ is the set of all sentences related to feature $\mathrm{f_j}, 1 \le \mathrm{j} \le \mathrm{m}$, the distance between *PP* and *PQ* is the average distance, calculated as

$$\mathrm{dist}(\mathrm{PP}, \mathrm{PQ}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{SD}(\mathrm{v_{p,ij}}, \mathrm{v_q})}{\mathrm{m} \times \mathrm{n}},$$

where $\mathrm{v_{p,ij}}$ is the value of the pair with feature *f*, which appears in sentence *j* and review *i*. *SD*( ) is the semantic difference calculated by the traversed distance in the sematic network [36].

Additionally, SentiWordNet [35] is similar to WordNet but focuses on the orientation of opinions. It is the annotation of all synsets of WordNet according to the notions of positivity, negativity and neutrality. In the proposed framework, positivity and negativity scores (i.e. Pos() and Neg()) is specifically applied to subjective words in a product review or customer query text for estimating their opinion orientation scores. For example, the distance between ("Battery life", "satisfactory") and ("Battery life", "excellent") can be calculated accurately with SentiWordNet. A straightforward way to calculate the semantic distance of opinion orientation is:

$$\mathrm{SD}(\mathrm{v_{p,ij}}, \mathrm{v_q}) = \left| \left| pos(v_{p,ij}) - neg(v_{p,ij}) \right| - \left| pos(v_q) - neg(v_q) \right| \right|$$

iii) Distance between Usage Pairs

Coping with Usage Pairs in customer query is much different from Objective Pairs and Subjective Pairs since in rare cases a product specification mentions the most suitable usage scenario of the product, and only a few of product reviews actually provide comments on the usage. Hence collecting product usage information from product database or reviews corpus, in either a direct or indirect way, is quite difficult. It can be noticed that whether a product is suitable for a usage scenario depends on features it possesses. For example, a digital camera for "scuba diving" requires the camera to be with "wide ISO range", "watertightness" and "long battery life", etc., whereas a camera for "taking building" is normally associated with the features "wide angle" and "resolution", etc. In fact, these relationships can be captured and modeled by processing and analyzing existing reviews corpus and product databases. As such, the usage information of a product can be derived according to its associated features using some inference models.

The Naive Bayesian method is one of the most successful machine learning algorithms in the domain of intelligent data analysis. Despite the simplicity of Naïve Bayesian (NB), it is proven to be very effective [38]. In this research, based on the strong independence assumption [39], that is, the probability of each feature is independent of each other, we establish a Naïve Bayesian model to calculate the likelihood of a product being suitable for a "usage", denoted as $\Pr(\mathrm{U}|\mathrm{p}) = \Pr(\mathrm{U}|\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_\mathrm{k})$, where U is the dependent class variable with a number of different usage and $\mathrm{f}_\mathrm{i}, 1 \le \mathrm{i} \le \mathrm{k}$ is a variable representing the feature pertaining to a product p (to facilitate computation, all feature variables need to be unified to either categorical or numerical format in advance). The probability of product p being suitable for U, according to Naïve Bayesian method, can be calculated by:

$$\Pr(U|p) = \frac{1}{Z}\Pr(U)\prod_{i=1}^{k}\Pr(f_i|U),$$

where Z is the scaling factor depending on the $\Pr(\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_\mathrm{k})$ only.

A model learning stage is needed for estimating various terms of $\Pr(\mathrm{u}_\mathrm{j})$ and $\Pr(\mathrm{f}_\mathrm{i}|\mathrm{u}_\mathrm{j})$. In this stage, a training dataset is generated by processing the reviews corpus and product database, and then extracting products associated with usage information. In this research, $\Pr(\mathrm{u}_\mathrm{j})$ is estimated based on the fraction of products with usage u_j over the entire training dataset. Given usage u_j, similarly, $\Pr(\mathrm{f}_\mathrm{i}|\mathrm{u}_\mathrm{j})$ can be calculated based on the fraction with feature f_i over all products with usage u_j.

Let $\mathrm{PP} = (\mathrm{u}, \mathrm{v}_\mathrm{p})$ be a Usage Pair about a product p. Its feature value v_p can be quantified by the likelihood that product p is suitable for usage u, i.e. $\mathrm{v}_\mathrm{p} = \Pr(\mathrm{U} = \mathrm{u}|\mathrm{p})$. Assuming $\mathrm{PQ} = (\mathrm{u}, 1)$ is a Usage Pair from a query q sharing the same product feature f with PP. The distance between PP and PQ is the distance between v_p and 1, namely, $\mathrm{dist}(\mathrm{PP}, \mathrm{PQ}) = 1\text{-}\mathrm{v}_\mathrm{p}$.

iv) Synthesizing Distance

Notably, various distance measures normally distribute in different scales and, in addition, distance values derived by different distance measures may have distinctive amplitude scales and baselines. For example, v_p defined above is a probabilistic value, normally very small, and thus the distance between two Usage Pairs $(1 - v_p)$ has very narrow amplitude with baseline close to 1. On the other hand, the distance between two subjective pairs may have a much larger baseline since the average of the traversed distance between two concepts in the semantic network is used to estimate the distance. As such, directly synthesizing different types of distances, without normalization, is problematic because components with small distance values may be overwhelmed by those with much larger distance values. A linear transformation is defined by combining a Z-score [40] and a MAX-MIN standardization is used in this framework; the former converts the value scales to the same range about zero and the latter transforms the distance value into interval [0,1].

Let $\mathrm{d} = \mathrm{dist}(\mathrm{PP}, \mathrm{PQ})$ be the distance between two pairs PP and PQ which can be two objective, subjective or usage pairs, and assume D is the value domain of d. Then we have:

DEFINITION (Z-score Mapping). The *Z-score mapping* is denoted as $\mathrm{Z}: \mathrm{D} \to \mathrm{D}$, and for $d \in D$, $Z(d) = \frac{(d-\overline{D})}{\delta_D}$, where $\overline{D}$ and δ_D are average and standard deviation of D, respectively.

DEFINITION (MAX-MIN Mapping). The *MAX-MIN mapping* is denoted as $M: D \to D$. For any $d \in D$, $M(d) = \frac{d-MIN(D)}{MAX(D)-Min(D)}$, where $MAX(D)$ and $MIN(D)$ are the maximum and minimum of all $d \in D$, respectively.

Normalization Mapping, the composite of Z-score and Max-Min mappings, can be defined as:

DEFINITION (Normalization Mapping). The *Normalization mapping* is denoted as $\mathbb{N}: D \to D$, For any $d \in D$, $\mathbb{N}(d) = M\big(Z(d)\big)$.

By introducing the Normalization Mapping on the value domain of the same type of distance, baselines of different distance definitions are standardized and their respective value scales are unified into the interval [0,1] in order to facilitate synthesizing the overall distance. This Normalization Mapping is used in Section V for presenting the overall similarity metrics.

CASE-BASED RECOMMENDATION

The traditional Collaborative filtering method is known to be very effective in making recommendations. However, cold-starts and sparsity problems are the major obstacles [5]. These are serious problems, particularly in niche markets where users are very unlikely to have rated many items. Case-based reasoning (CBR) is one of the most successful machine learning approaches to solve new problems by retrieving and adopting solutions for similar old cases [41]. CBR system is based on a repository of cases (the case base) which constitute the expertise used for solving the past problems. New problems can be solved by searching for old cases similar to the new case and hence their solutions can be adopted to solve the new problem. CBR methodology normally involves four key steps: (1) retrieve the most similar cases by comparing past cases; (2) reuse the solution associated with the matched case for solving the current problem; (3) revise the new solution if necessary, and (4),

retain it in the case base. Case-based recommender systems (CBRS) use CBR methodology for recommendations generation, in which products are viewed as cases and encoded and saved in the case base. As such, recommendations can be retrieved from the case base by searching for cases analogous to the product described in the customer request [42]. The "Alignment Assumption"[43] of CBR allows products/cases to be compared based on their features, making CBRS a promising solution to feature-based shopping guiding system. CBRS have also proven to be able to start with even a small case base [42] and, therefore, have become important alternatives to CF in many application domains.

i) The Similarity Metrics

Given F= $\{f_1, f_2, \dots, f_m\}$, the set of all available features in the proposed recommendation model, a product case p defined on a subset of F is a vector of feature-value pairs, denoted as $p = [PP_1, \dots, PP_{kp}]$, where $PP_j = (f_{jp}, v_{jp}), 1 \leq j \leq k_p$, is a product feature-value pair (objective, subjective or usage). We also say that the pair PP_j appears in case p, denoted as $PP_j \prec p$.

Likewise, a customer query can be represented by $q = [PQ_1, \dots, PQ_{kq}]$, where $PQ_j = (f_{jq}, v_{jq}), 1 \leq j \leq k_q$ is a request feature-value pair (objective, subjective or usage), we say that the pair PQ_j appears in query q, denoted as $PQ_j \prec q$.

The key step in case-based recommendation is to compare the distance between the query (normally referred to as "new case" in CBR) and the products (the "solution cases"). Hence definition of the similarity metrics is required. In real application, it is possible that some feature information required by the customer is missing in the product KB, and vice versa. Therefore, only features involved in both the user request and the product KB are considered when calculating the overall distance. Hence given query q and product case p, their *overlapping feature set* is defined as: $F_{p,q} = \{f_j | PP_j = (f_j, v_{jp}), PP_j \prec p \text{ and } PQ_j = (f_j, v_{jq}), PQ_j \prec q\}$. The overall distance between p and q can by calculated by the Weighted Euclidean Distance measure:

$$DIST(p,q) = \sqrt{\sum_{f \in F_{p,q}} w_i \cdot \mathbb{N}(dist(PP, PQ))}$$

where $PP = (f, v_p) \prec p$ and $PQ = (f, v_q) \prec q$. w_i is the corresponding weight on the distance element and w_i satisfies $\sum_{i=1}^{|F_{p,q}|} w_i = 1$. In practice, important features can be highlighted by using a comparatively larger weight value. The similarity metric between p and q can be simply calculated as $SIM(p,q) = 1 - DIST(p,q)$, since the overall distance has been standardized into the interval [0,1] already.

ii) Matching Process based on Case-based Recommendation

The proposed framework follows problem-solving methodology similar to that used in case-based reasoning. First, a customer looking for a product is allowed to describe requirements in natural language. Depending on depth of knowledge of the domain, description of features in query text can be in objective terms, subjective terms, usage terms, or a mixture of them. For example, if a user inputs a query text "a digital camera with big LCD and at least 8 megapixels for travel", the text parser in the NLP module then identifies the involved features (i.e. "megapixel", "LCD", "travel") and their corresponding values (i.e. "8", "big", and "1", where the last is an implicit value). Ontologies are also involved in this process to deal with the synonyms. The NLP module finally encodes the query text into a new case so as to retrieve similar old cases from the case base. Three major steps are involved in this process, i.e. the input, product features retrieval, and the output.

Figure 2 illustrates the process of our case-based recommendation based on the original CBR cycle proposed in Aamodt (1994), which has been widely used in many CBR systems [43]. In this framework, the recommendation process starts with the query text input by the user, and the top-N most similar cases/products are presented after searching in the product KB, sorted by ranking scores (retrieve and reuse). The ranking process is dialog-driven, which allows the customer to interactively refine the query when the results are not satisfactory or too many recommendations are generated. Namely, the system presents a number of additional attributes related to the product category so that the user can narrow down the scope by specifying more accurate requirements. For example, if the customer finds that the recommended products are not what he/she really wants, the usage request "travel" can then be used to derive a group of associated features, i.e. "weight", "size" and "battery life", and thus the customer may realize that what is actually needed is "long battery life" while there is no demand on the other two features. The aforementioned Naïve Bayesian underpinned by ontologies enables this inference from the usage request to product objective features. When the customer is pleased with the recommendations, choices are saved and utilized to refine the case base (revise, review and retain). For example, the Naïve Bayesian model can be adjusted by updating the probability $\Pr(f_i | u_j)$, where f_i and u_j are the feature "battery life" and usage "travel" respectively.

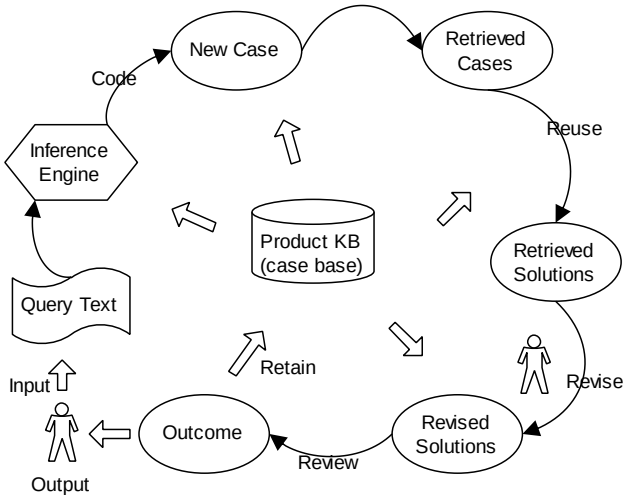

Figure 2. The process of case-based recommendation

SUMMARY AND FUTURE RESEARCH

This research introduces a novel framework for product recommendation in accordance with customer requests in natural language. It exploits NLP techniques and product opinion mining approaches to generate the product knowledge base. Several ontologies defined on the product

domain are constructed in order to support the inferences between terms with similar semantics. A case-based recommendation approach is used in this framework in order to avoid the cold-start problem in traditional collaborative filtering method. The similarity metrics used in case-based recommendation process is also elaborated. The recommender system based on the proposed framework can serve as intelligent guidance for online shoppers, especially for those without adequate domain expertise. This system will be implemented and experiments will be conducted in order to empirically evaluate the effectiveness of the framework and the associated methods proposed in this paper.

ACKNOWLEDGEMENT

This research is fully supported by research grant from University of Macau number RG014/09-10S/THA/FBA.

REFERENCES

[1] Wolfinbarger, M. and M. Gilly. Consumer motivations for online shopping. in Americas Conference on Information Systems. 2000. Long Beach, California.

[2] Scaffidi, C., et al. Red Opal: product-feature scoring from reviews. in the 8th ACM conference on Electronic commerce. 2007. San Diego: ACM.

[3] Ricci, F., et al., Recommender systems handbook. 2011: Springer.

[4] Adomavicius, G. and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 2005: p. 734-749.

[5] Schein, A.I., et al. Methods and Metrics for Cold-Start Recommendations. in 25th Annual International ACM SIGIR 2002. Tampere, Finland.

[6] Herlocker, J.L., J.A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. in ACM conference on Computer supported cooperative work. 2000. Philadelphia: ACM.

[7] Horrigan, J.A., Online shopping. Pew Internet & American Life Project Report, 2008. 36.

[8] Shardanand, U. and P. Maes. Social information filtering: algorithms for automating "word of mouth". in SIGCHI conference on Human factors in computing systems. 1995. Denver, Colorado: ACM Press.

[9] Senecal, S. and J. Nantel, The influence of online product recommendations on consumers' online choices. Journal of Retailing, 2004. 80(2): p. 159-169.

[10] Cazella, S.C. and L.O.C. Alvares, Combining Data Mining Technique and Users' Relevance Opinion to Build an Efficient Recommender System. Revista Tecnologia da Informa o, UCB, 2005. 4(2).

[11] Sun, J., et al., Mining Reviews for Product Comparison and Recommendation. Research Journal on Computer Science and Computer Engineering with Applications, 2009(39): p. 33-40.

[12] Linden, G., B. Smith, and J. York, Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 2003. 7(1): p. 76-80.

[13] Aciar, S., et al. Recommender system based on consumer product reviews. in IEEE/WIC/ACM International Conference on Web Intelligence. 2006. Hong Kong: IEEE Computer Society.

[14] Dellarocas, C., The digitization of Word-of-Mouth: Promise and challenges of online feedback mechanisms. Management Science, 2003. 49(10): p. 1407-1424.

[15] Dellarocas, C., Strategic manipulation of Internet opinionforums: Implications for consumers and firms. Management Science, 2006. 52(10): p. 1577-1593.

[16] Popescu, A.M. and O. Etzioni. Extracting product features and opinions from reviews. in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. 2005. Stroudsburg, PA.

[17] Pang, B. and L. Lee, Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, 2008. 2(1-2): p. 1-135.

[18] Ku, L.W. and H.H. Chen, Mining opinions from the Web: Beyond relevance retrieval. Journal of the American Society for Information Science and Technology, 2007. 58(12): p. 1838-1850.

[19] Charniak, E., Statistical techniques for natural language parsing. AI Magazine, 1997. 18(4): p. 33.

[20] Turney, P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. in the Meeting of the Association for Computational Linguistics (ACL). 2002.

[21] Jindal, N. and B. Liu. Review Spam Detection. in the16th International World Wide Web Conference. 2007. Banff, Canada.

[22] Mukherjee, A., et al. Detecting Group Review Spam. in the 20th International World Wide Web Conference. 2011. Hyderabad, India.

[23] Dey, L. and S.K.M. Haque, Opinion Mining from Noisy Text Data. International Journal on Document Analysis and Recognition, 2009. 12(3): p. 205-226.

[24] Ott, M., et al. Finding deceptive opinion spam by any stretch of the imagination. in 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. Portland, Oregon.

[25] Wiebe, J.M. Learning subjective adjectives from corpora. in AAAI. 2000. Cambridge, MA.

[26] Zhuang, L., F. Jing, and X.Y. Zhu. Movie review mining and summarization. in 15th ACM international conference on Information and knowledge management. 2006. New York, NY: ACM.

[27] Loper, E. and S. Bird. NLTK: The Natural Language Toolkit. in the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. 2002. Philadelphia.

[28] Cunningham, H., et al. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. in the 40th Anniversary Meeting of the Association for Computational Linguistics. 2002. Philadelphia.

[29] Gruber, T., Collaborating around Shared Content on the WWW, in W3C Workshop on WWW and Collaboration. 1995: Cambridge, MA.

[30] Cheng, C.K., X. Pan, and F. Kurfess, Ontology-based semantic classification of unstructured documents, in 1st International Workshop on Adaptive Multimedia Retrieval. 2003: Hamburg, Germany. p. 441-458.
[31] Zhou, L. and P. Chaovalit, Ontology supported polarity mining. Journal of the American Society for Information Science and Technology, 2008. 59(1): p. 98-110.
[32] Lau, R.Y.K., et al. Automatic Domain Ontology Extraction for Context-Sensitive Opinion Mining. in the 13th International Conference on Information Systems. 2009. Phoenix, Arizona.
[33] Maedche, A. and S. Staab. Semi-Automatic Engineering of Ontologies from Text. in 12th International Conference on Software Engineering and Knowledge Engineering. 2000. --.
[34] Miller, G.A., et al., Introduction to wordnet: An on-line lexical database. International Journal of lexicography, 1990. 3(4): p. 235.
[35] Baccianella, S., A. Esuli, and F. Sebastiani, SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in Language Resources and Evaluation. 2010: Valletta, Malta.
[36] Roddick, J.F., K. Hornsby, and D. de Vries. A Unifying Semantic Distance Model for Determining the Similarity of Attribute Values. in 26th Australasian Computer Science Conference. 2003. Adelaide, Australia.
[37] Patwardhan and Pedersen. Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts. in EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together. 2006. Trento, Italy.
[38] Zhang, H., The Optimality of Naive Bayes, in 17th Florida Artificial Intelligence Research Society Conference. 2004: Miami Beach, Florida.
[39] Cheeseman, P. and J. Stutz, Bayesian Classification (AutoClass): Theory and Results, in Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, et al., Editors. 1996, AAAI Press/MIT Press.
[40] Han, J. and M. Kamber, Data Mining: Concepts and Techniques. 2nd ed. 2006: Morgan Kaufmann.
[41] Riesbeck, C.K., et al., Inside case-based reasoning. 1989: L. Erlbaum Associates Inc.
[42] Lorenzi, F. and F. Ricci, Case-based recommender systems: A unifying view. Intelligent Techniques for Web Personalization, 2005: p. 89-113.
[43] Leake, D.B., CBR in context: The present and future, in Case based reasoning: Experiences, lessons, and future directions, D.B. Leake, Editor. 1996, MIT, Cambridge. p. 3-30.